

Experience with the Architecture Improvement Workshop

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Larry Jones, SEI
Rick Kazman, SEI
SATURN Conference, May 7, 2009



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

I've Evaluated My Architecture. Now What?

You sold me on doing an ATAM evaluation
of my architecture.

The "Architecture Paratroopers"

- landed
- did an ATAM evaluation
- gave me a list of risks
- pulled out

What do I do with this information?

*Answer: Conduct an Architecture
Improvement Workshop.*



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

2



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

1

Report Documentation Page			<i>Form Approved OMB No. 0704-0188</i>	
<p>Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p>				
1. REPORT DATE 07 MAY 2009	2. REPORT TYPE	3. DATES COVERED 00-00-2009 to 00-00-2009		
4. TITLE AND SUBTITLE Experience with the Architecture Improvement Workshop		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Carnege Mellon University, Software Engineering Institute, Pittsburgh, PA, 15213		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S)		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited				
13. SUPPLEMENTARY NOTES presented at Software Engineering Institute (SEI) Architecture Technology User Network (SATURN) Conference, 4-7 May 2009, Pittsburgh, PA.				
14. ABSTRACT				
15. SUBJECT TERMS				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 22
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified	19a. NAME OF RESPONSIBLE PERSON	

Agenda

- The ATAM and its Outputs
- The Architecture Improvement Workshop
- Some Experiences with the Architecture Improvement Workshop
- Extensions to the Architecture Improvement Workshop
- Conclusion



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

3

Agenda

► ***The ATAM and its Outputs***

- The Architecture Improvement Workshop
- Some Experiences with the Architecture Improvement Workshop
- Extensions to the Architecture Improvement Workshop
- Conclusion



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

4



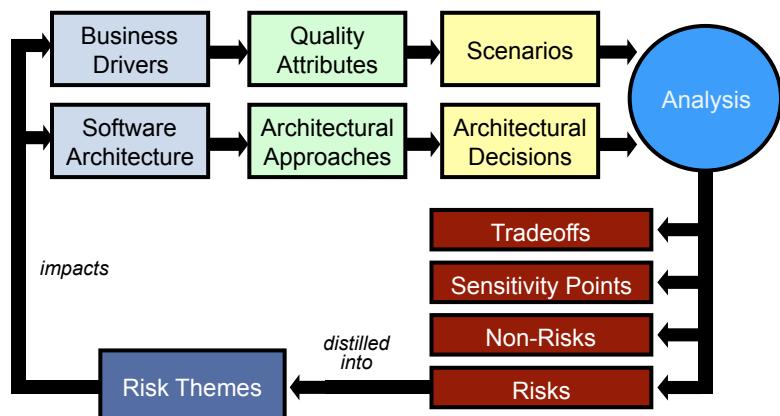
Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

2

Conceptual Flow of the ATAM



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

5

Outputs of an ATAM Evaluation

The outputs of an ATAM evaluation include

- a utility tree
 - a hierarchical view of important system qualities including operational definitions of what these qualities mean
- a set of stakeholder-generated scenarios
 - describing how the architecture responds to important system uses, including those that stretch or expand the design space
- sets of risks, non-risks, tradeoffs, and sensitivity points
- a set of risk themes that summarize the risks and their impact the system's mission or business drivers
- better communications among stakeholders
- (often) better software architecture documentation

These serve as key inputs to the Architecture Improvement Workshop.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

6



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

3

Agenda

The ATAM and its Outputs

The Architecture Improvement Workshop

Some Experiences with the Architecture Improvement Workshop

Extensions to the Architecture Improvement Workshop

Conclusion



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

7

Comparison of AIW and E²AIW

**Focus: Improve
architecture in near term**

AIW

1. Review *business goals*.
2. Review *utility tree, scenarios*.
3. Complete the analysis of any *critical scenarios*.
4. Brainstorm *architectural strategies*.
5. Analyze proposed *architectural strategies*.

**Focus: Position architecture for
improvement in a less-certain future**

E²AIW

1. Characterize *business goals* and constraints.
2. Characterize *quality attribute scenarios*
3. Characterize *quality attribute scenario utility* and *uncertainty*
4. Brainstorm *architectural options*.
5. Associate an "Expected Value-for-Cost" measure with *architectural options*.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

8



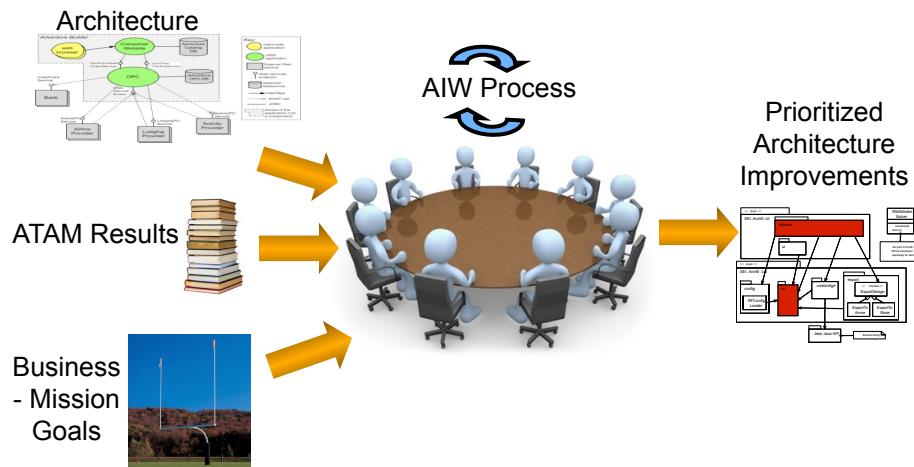
Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

4

Overview of the Architecture Improvement Workshop (AIW)



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

9

Conducting an AIW

Participants

- architect
- additional key technical and managerial stakeholders (just a few)
- facilitator, recorder

Steps

1. Review business goals
2. Review utility tree and scenarios.
3. Complete the analysis of critical scenarios.
4. Brainstorm architectural strategies.
5. Analyze proposed architectural strategies.

Can package into about 2 days

- Steps 1-4 conducted in one session
- (optional break for off-line confirmation of priorities, investigation of details)
- Step 5 conducted in one session



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

10



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

5

Steps 1 – 3: Review, Refine Key ATAM work

1. Review Business Goals
2. Review Utility Tree, Scenarios
3. Complete the analysis of any critical scenarios

Didn't I already do this during the ATAM?

Yes, but

- It may have been awhile since the ATAM.
- Sometimes in the rush to complete the ATAM some things may be less polished than they need to be for more detailed work.

Participants should do this review prior to the AIW and come prepared to refine as necessary.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

11

Step 4: Brainstorm Architectural Strategies

- i. Review the high priority scenarios from the ATAM. With these in mind brainstorm a list of architectural strategy options that might address issues in one or more of the scenarios.
- ii. Cluster, refine, and prioritize the options.
- iii. For each option record:
 - an identifying name, a high level description, business drivers, applicable scenarios, qualities and concerns
 - postulate a set of tactical decisions to address the option
 - identify the components impacted and how
 - identify any constraints, assumptions, and derived requirements
 - defer more detailed analysis.

ADVICE: Don't get bogged down on identifying details. Make an action item list to identify any necessary off-line work.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

12



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

6

Off-line Work Between Sessions

Address any action items.

Complete option details, particularly

- scenarios affected by the design options
- impact on the scenario if the design option is implemented

Confirm or adjust priorities.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

13

Step 5: Analyze Proposed Architectural Strategies

Discuss each architectural strategy option in priority order in detail.

Make rough effort/cost estimates if possible to help prioritize.

ADVICE:

- While an in-depth discussion is appropriate, again use an action item list to avoid bogging down.
- Remember the purpose is to provide sufficient insight to the architect and managers to proceed.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

14



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

7

Agenda

The ATAM and its Outputs

The Architecture Improvement Workshop

Some Experiences with the Architecture Improvement Workshop

Extensions to the Architecture Improvement Workshop

Conclusion



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

15



SSNAM: Space Surveillance Network Analysis Model - Overview

SSNAM is

- part of a space-sensor force-structure decision process
- uses a high-fidelity simulation model to assess operational impact of changes to the space sensor network (SSN)
- uses operational code to produce credible results

SSNAM is very complex to set up and run.

- takes 16-18 hours to run in a lower fidelity mode on a 16-PC cluster
- takes 4-5 days to run in high fidelity mode on a 16-PC cluster
 - or about 1 day on the MHPCC (Maui High Performance Computing Center) 50-node Hoku cluster

Business / mission goals for upgrading SSNAM

- increasing the ease of use
- making it easier for government personnel to run the system without contractor support
- improve understandability of the output
- decreasing run-time
- decreasing time to incorporate new sensor models



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

16



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

8

SSNAM Improvement Examples - 1

Many high-priority scenarios pointed to reliability problems. Failures could occur at a number of points and cause the loss of an entire run.

The process led to the design option named “check-point restore framework.”

Analysis revealed that in addition to comprehensive changes, there were several opportunities for immediate, inexpensive improvements (“low hanging fruit”).

Improved heartbeat monitoring and status polling was implemented.

Results

- Better status reporting has led to earlier error detection and recovery, and better machine utilization.
- Most failures (typically environmental) that are out of the control of the SSNAM team have been mitigated.
- Failure recovery to the previous simulated day is now possible without the performance impact of check-pointing



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

17

SSNAM Improvement Examples - 2

Other architecture improvements were identified along with cost estimates and were prioritized for future implementation. These included architectural restructuring to

- remove redundant computations
- add front-end data preparation tools to reduce busted runs
- provide better back-end analysis tools
- build generic sensor models to allow more rapid network composition analysis

Bottom Line:

- The Air Force Space Command customer, the supporting contractor, and their collaborators at the High Performance Computing Software Applications Institute for Space Situational Awareness (HSAI-SSA) were all very happy with the process and the results.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

18



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

9

Agenda

- The ATAM and its Outputs
- The Architecture Improvement Workshop
- Some Experiences with the Architecture Improvement Workshop
- Extensions to the Architecture Improvement Workshop***
- Conclusion



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

19

Comparison of AIW and E²AIW

**Focus: Improve
architecture in near term**

AIW

1. Review *business goals*.
2. Review *utility tree, scenarios*.
3. Complete the analysis of any *critical scenarios*.
4. Brainstorm *architectural strategies*.
5. Analyze proposed *architectural strategies*.

**Focus: Position architecture for
improvement in a less-certain future**

E²AIW

1. Characterize *business goals* and constraints.
2. Characterize *quality attribute scenarios*
3. Characterize *quality attribute scenario utility* and *uncertainty*
4. Brainstorm *architectural options*.
5. Associate an "Expected Value-for-Cost" measure with *architectural options*.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

20



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

10

Extended Evolutionary AIW (E²AIW)

We are now working on E²AIW: An Extended Evolutionary Architecture Improvement Workshop

An E²AIW considers both evolution (multiple future states), as well as multiple possible characterizations of each state, according to the *uncertainty* surrounding each.

In response to these future states the E²AIW proposes *architectural options* and a way of characterizing the costs, benefits, and uncertainty of each.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

21

E²AIW Steps

1. Characterize Business Goals and Constraints
2. Characterize Quality Attribute Scenarios
3. Characterize Quality Attribute Scenario Utility and Uncertainty
4. Brainstorm Architectural Options
5. Associate an “Expected Value-for-Cost” measure with Architectural Options



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

22



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

11

1. Characterize Business Goals and Constraints

What is the future value of moving to a more flexible architecture today? Can it help to contain future costs, for example?

Some possible future states to consider:

- increase in number of users and new types of users?
- increase in number of information resources and types?
- increase in number of new types of systems and interactions among systems? New missions? Modifications to missions from doctrinal changes?



Software Engineering Institute

Carnegie Mellon

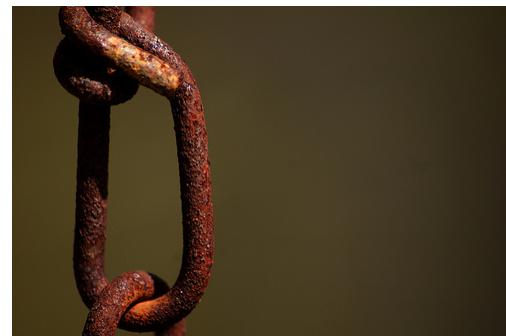
Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

23

1. Characterize Business Goals and Constraints

Delineate the constraints under which the proposed evolution will take place, for example:

- i. COTS
- ii. Budget
- iii. Schedule
- iv. Legacy systems
- v. Legacy interfaces



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

24



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

12

2. Characterize Quality Attribute Scenarios

Conduct an *extended* QAW to

- i. understand and characterize the base state in terms of QAs
- ii. brainstorm scenarios representing future states: interoperability, performance, portability, availability, security scenarios, etc.
- iii. group similar/related scenarios
- iv. attempt to enumerate future states in terms of groups of the most important QA requirements

An extended QAW collects multiple response goals for each scenario.



Software Engineering Institute

Carnegie Mellon

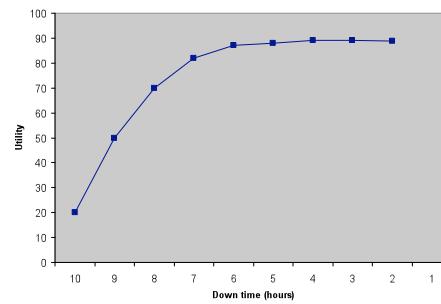
Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

25

3. Characterize Quality Attribute Scenario Utility and Uncertainty - 1

Associate “utility” with the current state and with the requirements of future states in terms of response goals, e.g.

- how useful it is to accommodate a technology insertion in 6 person weeks? 4 weeks? 1 week?
- how useful is it to have an average latency in degraded communications mode of 2 seconds? 1 second? 0.5 seconds?



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

26



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

13

3. Characterize Quality Attribute Scenario Utility and Uncertainty - 2

Characterize the “envelope of uncertainty”.

- such a characterization will be accomplished using Likert scales; no attempt at precisely capturing uncertainty will be made at this stage.

Prioritize the scenarios according to expected Δutility



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

27

4. Brainstorm Architectural Options - 1

Consider *architectural options* that address the high priority QA scenarios representing future states (the “evolution” scenarios)

Determine the expected response levels of each architectural option with respect to the scenarios

- these “expected” levels may be determined by: estimation, guesses, analogy, analysis, prototyping/experiments, qualification/certification
- along with each determination, attempt to assess the degree of uncertainty of the estimate, even if this is crude (e.g. a guess is more uncertain than a paper-and-pencil analysis, which is more uncertain than prototyping/experiments)



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

28



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

14

4. Brainstorm Architectural Options - 2

Estimate the cost of each architectural option

Consider *portfolios* of architectural options, including a consideration of the side-effects of architectural options in terms of both costs and benefits



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

29

5. Associate an “Expected Value-for-Cost” measure with Architectural Options

For each option, the potential future states, along with their uncertainty, can be used to calculate an expected value.

For each option, estimate the cost, along with its envelope of uncertainty

- both up-front and ongoing costs will need to be elicited as these may distinguish among the architectural options.
- on-going costs may be split into base costs and the costs of exercising options.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

30



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

15

Make Decisions

Decisions (choosing a staged set of architectural options) can now be undertaken that take into account future states, response levels, costs, and uncertainty.

This will require iterating on steps 4-5 as architectural options will have dependencies that will influence their costs and benefits.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

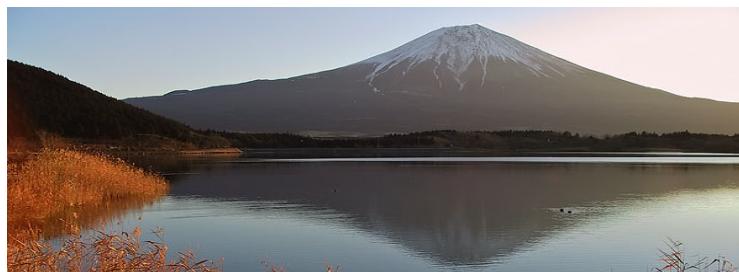
31

Reflection

The E²AIW is a “heavier” method than the AIW.

It must be so, for it is attempting to make an architecture “future-proof” and that is a complex task with many variables and much uncertainty.

However, it is grounded in proven existing techniques: the AIW, the CBAM, and the theory of real options.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

32



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

16

Status

The AIW is road-tested and can be applied anywhere that an ATAM can be applied.

The E²AIW is now in the process of moving from the lab to the field.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

33

Agenda

The ATAM and its Outputs

The Architecture Improvement Workshop

Some Experiences with the Architecture Improvement Workshop

Extensions to the Architecture Improvement Workshop

Conclusion



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

34



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

17

Comparison of AIW and E²AIW

Focus: Improve architecture in near term

AIW

1. Review *business goals*.
2. Review *utility tree, scenarios*.
3. Complete the analysis of any *critical scenarios*.
4. Brainstorm *architectural strategies*.
5. Analyze proposed *architectural strategies*.

Focus: Position architecture for improvement in a less-certain future

E²AIW

1. Characterize *business goals and constraints*.
2. Characterize *quality attribute scenarios*
3. Characterize *quality attribute scenario utility and uncertainty*
4. Brainstorm *architectural options*.
5. Associate an "Expected Value-for-Cost" measure with *architectural options*.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

35

Contact Information

Dr Larry Jones

Senior Member of the Technical Staff
Research, Technology, and System Solutions

Telephone: +1 719-548-4744
Email: ljj@sei.cmu.edu

Dr Rick Kazman

Senior Member of the Technical Staff
Research, Technology, and System Solutions
Telephone: +1 412-268-1588
Email: kazman@sei.cmu.edu

U.S. mail:

Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

World Wide Web:

www.sei.cmu.edu
www.sei.cmu.edu/contact.html

Customer Relations

Email: customer-relations@sei.cmu.edu
Telephone: +1 412-268-5800
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

36



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

18

Appendix



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

37

AIW Brainstorm Template

Option Name			
Business Goals Addressed			
Technical Description			
Scenarios Affected	Scenario	Attribute/Concern	Impact
			Current State:
			Option Impact:

Components Affected	Component	How Affected	
Constraints		...	
Assumptions			
Derived Architectural Requirements			



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

38



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

19

AIW Analysis Template

Option Name			
Business Goals Addressed			
Technical Description			
Scenarios Affected	Scenario	Attribute/Concern	Impact
			Current State:
			Option Impact:
			...
			...
Components Affected	Component		How Affected
Constraints			
Assumptions			
Derived Architectural Requirements			
Testing Required			
Risks			
Est. Cost			
Est. Time to Complete			



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

39

Utility Curve Elicitation Template

Scenario	Votes	Utility Scores for Response Measure Goals				
			Worst	Current	Desired	Best
1		Response Measure Goal				
		Utility				
2		Response Measure Goal				
		Utility				
3		Response Measure Goal				
		Utility				
4		Response Measure Goal				
		Utility				



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

40



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

20

Expected Response / Architecture Strategy Template

AS	AS Name	Scenarios Affected	Current Response	Expected Response
1		1		
		2		
		3		
		4		
2		1		
		2		
		3		
		4		
3		1		
		2		
		3		
		4		



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

41

Benefit Calculation Template

AS	Scenario	Benefit $\Delta \text{Utility} = \text{Utility}_{\text{expected}} - \text{Utility}_{\text{current}}$	Votes	Normalized Benefit (Benefit x Votes)	Total Benefit $\sum \text{Scenario} \text{ Normalized Benefit}$
1	1				
	2				
	3				
	4				
2	1				
	2				
	3				
	4				
3	1				
	2				
	3				
	4				



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

42



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

21

NO WARRANTY

THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.



Software Engineering Institute

Carnegie Mellon

Experience with the AIW
Jones, Kazman, SATURN, May 2009
© 2009 Carnegie Mellon University

43



Software Engineering Institute

Carnegie Mellon

© 2009 Carnegie Mellon University

22